

title: React Typescript 打包方案

date: 2019-07-15

tag: Babel, React

## description: 提供 React + Typescript 代码打包的两种方案。

最近用 React + Typescript 写了 SVG 编译成 Inline SVG (React SVG) 的应用。通过 `tsc` 命令打包成 `*.js` 文件，然后在外部引用。对打包可用性进行，通过使用以及测试引入验证。

## 使用

```
{
  "scripts": {
    "wizard-icons": "git+https://github.com/xxx-icons.git"
  }
}
```

引入以及测试

```
// index.js 文件引用
import icons from 'xxx-icons';
// index.test.js UT 测试
import React from 'react';
import Icon from './index';
import { render } from 'enzyme';

describe('Icon', () => {
  it('should render default ui', () => {
    const icon = render(<Icon type="xxx-icon"/>);
    expect(icon).toMatchSnapshot();
  });
})
```

## tsc 命令打包

打包的代码

```
{
  "scripts" : {
    "ts2js": "tsc --p ./lib-tsconfig.json"
  }
}
```

lib-tsconfig.json 配置

```
{
  "compilerOptions": {
    "outDir": "./lib/",
    "module": "commonjs",
    "declaration": true,
    "target": "es5",
    "jsx": "react",
    "moduleResolution": "node",
    "allowSyntheticDefaultImports": true,
    "forceConsistentCasingInFileNames": true,
    "suppressImplicitAnyIndexErrors": true,
    "strict": true
  }
}
```

可以得到 \*.js 和 \*.d.ts 文件。打包文件代码

```
// *.js
"use strict";
var __assign = (this && this.__assign) || function () {
  __assign = Object.assign || function(t) {
    for (var s, i = 1, n = arguments.length; i < n; i++) {
      s = arguments[i];
      for (var p in s) if (Object.prototype.hasOwnProperty.call(s, p))
        t[p] = s[p];
    }
    return t;
  };
  return __assign.apply(this, arguments);
};
Object.defineProperty(exports, "__esModule", { value: true });
var react_1 = require("react");
var SvgCycle = function (props) { return (react_1.default.createElement("svg", __assign({ viewBox: "0 0 32 32", width: "1em", height: "1em", fill: "none", stroke: "black", stroke-width: "1px" }, props), react_1.default.createElement("path", { d: "M12.244 2.922L11.648 1C5.214 2.898.5 8.88.5 15.958c0 4.766 2.16 9.012 5.522 12.244L11.648 22.078c1.648 1.648 4.766 2.922 8.88 2.922c4.766 0 8.88-4.766 8.88-8.88c0-4.766-2.16-9.012-5.522-12.244L11.648 2.922Z" }));
exports.default = SvgCycle;
// *.d.ts
import React from "react";
declare const SvgClose: (props: React.SVGProps<SVGSVGElement>) => JSX.Element;
export default SvgClose;
```

## Babel 命令打包

### 安装依赖包

```
yarn add --dev \
  @babel/core \
  @babel/cli \
  @babel/preset-env \
  @babel/preset-typescript \
  typescript
```

### 配置文件

babel.config.js 配置

```
module.exports = {
  presets: ['@babel/preset-typescript', '@babel/preset-env']
}
```

### 命令

```
{
  "scripts" : {
    "babel src --out-dir lib --extensions '.ts,.tsx'",
  }
}
```

### 打包代码

```
// *.js
"use strict";

Object.defineProperty(exports, "__esModule", {
  value: true
});
exports["default"] = void 0;
```

```
var _react = _interopRequireDefault(require("react"));

function _interopRequireDefault(obj) { return obj && obj.__esModule ? obj : { "default": obj }; }

function _extends() { _extends = Object.assign || function (target) { for (var i = 1; i < arguments.length; i++) { var sou

var SvgClose = function SvgClose(props) {
  return _react["default"].createElement("svg", _extends({
    viewBox: "0 0 32 32",
    width: "1em",
    height: "1em"
  }, props), _react["default"].createElement("path", {
    d: "M30 3.346L28.654 2 16 14.654 3.346 2 2 3.346 14.654 16 2 28.654 3.346 30 16 17.346 28.654 30 30 28.654 17.346 16z"
  }));
};

var _default = SvgClose;
exports["default"] = _default;
```

## 注意

babel 没有打包 \*.d.js 文件，所以在做 TS 开发时需要声明

```
// global.d.ts
declare module 'xxx-icons';
```